Transport Level Security: a proof using the Gong–Needham–Yahalom Logic

Walter D Eaves

February 1, 2008

Abstract

This paper provides a proof of the proposed Internet standard Transport Level Security protocol using the Gong–Needham–Yahalom logic. It is intended as a teaching aid and hopes to show to students: the potency of a formal method for protocol design; some of the subtleties of authenticating parties on a network where all messages can be intercepted; the design of what should be a widely accepted standard.

1 Transport Level Security Protocol

This section provides an insight into the workings of the next generation of authentication protocol: the Transport Level Security Protocol version 1.0[DA97], the successor to the Secure Sockets Layer[FKK95]. To do this, the Gong–Needham–Yahalom, GNY, logic [GNY90] is introduced which is a formal method for proving the safety of a cryptographically-based protocol. It is described at length in appendix A. When working through protocols the relevant rule of inference will be stated and will refer to those in the appendix.

The *Transport Level Security handshake protocol*[DA97], TLS, has an unknown heritage, but it has a great deal of similarity to that described in [DS81]. It is predicated on the existence of readily available public keys: TLS's predecessor made use of X.509 certificates, see [CCI88], issued by a Certification Authority, CA, an example of which is *Thawte*[THA99]. A discussion of the limitations of certificate technology can be found in Röscheisen's on–line paper [Ros95].

TLS has three sub–protocols:

- Server anonymous
- Server named, client anonymous
- Server named, client named

These differ by who is required to send their X.509 certificates, the key exchange protocol is different only when the client is named and thus has a public–key that can be used. The messages are shown in figure 1 sent during a run of the protocol are more or less the same for all sub–protocols. As can be seen, no key issuing server is needed.

The TLS is a more complicated protocol than the *Kerberos* which is described in the appendix §A.3 , before looking at TLS's protocol proof, it might be best to examine *Kerberos's*. Also, there are some more examples of other authentication protocols being investigated and found lacking[Low96].

A protocol proof has three stages:

- Message Analysis
- Pre–conditions Analysis
- Belief deductions for each message

Message analysis involves formalizing the content of messages so that they contain just keys and identifiers. Pre–conditions analysis formalizes what the parties to the protocol assume about the state of keys before undertaking the protocol run. Belief deductions analyzes how each party can deduce new beliefs when it receives a new message.

2 Messages for the Named Server Protocol

There are six message exchanges. There is a provision for more, to settle which cryptographic implementations to use and for the client to provide a certificate, but this is the basic protocol for a named server to an anonymous client.

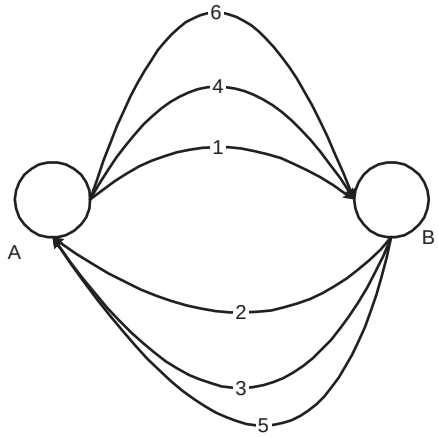

Figure 1: TLS protocol: Messages

Messages

$$A \to B\colon (N_A, T_A) \qquad (M_1)$$
$$B \to A\colon (N_B, T_B) \qquad (M_2)$$
$$B \to A\colon \{+K_B, \langle B\rangle, \langle C\rangle\}_{-K_C} \qquad (M_3)$$
$$A \to B\colon \{N'_A\}_{+K_B} \qquad (M_4)$$
$$B \to A\colon \{H(K_{AB}, AB_5, (M_1, M_2, M_3, M_4))\}_{K_{AB}} \qquad (M_5)$$
$$A \to B\colon \{H(K_{AB}, AB_6, (M_1, M_2, M_3, M_4))\}_{K_{AB}} \qquad (M_6)$$

where

$$K_{AB} = F((N_A, T_A, N_B, T_B), N'_A)$$

and, the behaviour of the public and private keys is:

$$\{\{X\}_{-K}\}_{+K} = X$$
$$\{\{X\}_{+K}\}_{-K} = X$$

and

$$AB_5 \triangleq \text{ “server finished” } , \; AB_6 \triangleq \text{ “client finished”}$$

The messages can be summarized as follows:

M_1 A sends a timestamp and a nonce to B.

M_2 B sends another timestamp and another nonce to A.

M_3 B sends its certificate signed by the certification authority; it contains $+K_B$, B's public–key.

M_4 A returns the “pre–master secret” N'_A encrypted under $+K_B$.

M_5 B sends a hash of the session key, a tag indicating the protocol stage AB_5[1] and all preceding messages exchanged to A.

M_6 A sends a hash of the session key, a tag indicating the protocol stage AB_6 key and all preceding messages to B.

3 Pre–Conditions

Certificates

1. Some Expectations

   A expects to use C as the certification authority and expects to use B, so

   $$A \ni \langle C\rangle \quad A \ni \langle B\rangle \qquad (1i)$$

[1] Actually all stages of the protocol are marked with a stage identifier, but it is not necessary to consider all of them.

2. Using Them

   The role of the unseen certification authority, C, is pivotal. Even though C has used the private key $-K_C$ to create the certificate, this key is not used again.

$$B \mid\equiv A \mid\equiv \overset{+K_C}{\rightarrow} C, \quad A \mid\equiv \overset{+K_C}{\rightarrow} C \tag{1ii}$$

   The two parties rely on the public–key being available. A must be able to get K_C:

$$A \ni +K_C, \quad B \mid\equiv A \ni +K_C \tag{1iii}$$

3. Trusting in them

   The pre–conditions regarding B's certificate are as follows. A and B both trust C to deliver the correct identity with the public key.

$$A \mid\equiv C \mid\Rightarrow (+K_P, \langle P \rangle), \quad B \mid\equiv C \mid\Rightarrow (+K_P, \langle P \rangle) \tag{1iv}$$

4. Meaning of the contents

   For A the assumption underlying a certificate is that the public–key is the public–key of the named party.

$$A \mid\equiv (+K_P, \langle P \rangle) \rightsquigarrow C \mid\equiv \overset{+K_P}{\rightarrow} P \tag{1v}$$

   And A believes C when C names a key:

$$A \mid\equiv C \mid\Rightarrow \overset{+K_P}{\rightarrow} P \tag{1vi}$$

System Capabilities

1. B believes that A can generate a nonce and keep it secret to pass it on as the pre–master secret.

$$B \mid\equiv A \mid\Rightarrow \#(X), \quad B \mid\equiv A \overset{X}{\leftrightarrow} A \tag{1vii}$$

2. A and B have both assumed that the other can generate the master secret if presented with the components.

$$A \mid\equiv B \mid\Rightarrow F(X, Y), \quad B \mid\equiv A \mid\Rightarrow F(X, Y) \tag{1viii}$$

   and, of course, they do

$$B \mid\Rightarrow F(X, Y), \quad A \mid\Rightarrow F(X, Y) \tag{1ix}$$

3. B has a private–key and holds his own certificate.

$$B \ni -K_B, \quad B \ni \{+K_B, \langle B\rangle, \langle C\rangle\}_{-K_C} \tag{1x}$$

Note that the format of a certificate does includes a statement of the identity of C. Although not used in this protocol it is an important part of it since it allows the public–key of C to be checked.

4 Belief Deductions Analysis

1. Messages M_1 received by B and M_2 received by A

   These nonce and timestamp exchanges are important, because they are used in the generation of the key. The vindication is the appearance of the time–stamp, which is definitely fresh, and the the rule (F1) freshens the nonces.

$$\begin{array}{lll} A \ni M_1, M_2 \text{ By (P1)} & & B \ni M_1, M_2 \text{ By (P1)} \\ A \ni N_B, T_B, N_A, T_A & \text{and} & B \ni N_A, T_A, N_B, T_B \\ A \mid\equiv \#\,(N_B) & & B \mid\equiv \#\,(N_A) \end{array} \tag{2i}$$

2. Message M_3 received by A

   A receives the certificate from, presumably, B. By (1ii), (1iii) and (I4), A now has:

$$A \mid\equiv C \mid\sim \{+K_B, \langle B\rangle, \langle C\rangle\}_{-K_C}$$

   and can decrypt the contents to discover:

$$C \mid\sim (+K_B, \langle B\rangle)$$

   By (1iv) and (1v)

$$A \mid\equiv C \mid\equiv \overset{+K_B}{\rightarrow} B$$

   By (1vi) and (J1)

$$A \mid\equiv \overset{+K_B}{\rightarrow} B$$

   And of course

$$\begin{array}{l} A \ni M_3 \text{ By (P1).} \\ A \ni M_4 \text{ Because } A \text{ creates it.} \end{array} \tag{2ii}$$

   Notice that A does not know that the sender of this message was B.

3. Message M_4 received by B

   B gains knowledge of the following:

$$\begin{aligned} &B \ni M_3 \text{ Since it sent it} \\ &B \ni M_4 \text{ By (P1)} \end{aligned} \tag{2iii}$$

   From which by (I2) where $\langle S \rangle$ is ${N'}_A$.

$$\begin{aligned} B &\lhd {N'}_A \\ B &\ni {N'}_A \end{aligned}$$

   This is something of an innovation: ${N'}_A$ has not been established as a shared secret by the conventional method of passing it along a channel secured by a long–term key or comparing it to a pre–stored hash, but it *will* be established as a secret by the subsequent correct operation of the protocol.

   Since ${N'}_A$ came under cover of the public key and it will be later identified as unique to the sender, it is a shared secret, so by (1vii).

$$\begin{aligned} B &\mid\equiv A \overset{{N'}_A}{\leftrightarrow} B \\ B &\mid\equiv \#\left({N'}_A\right) \end{aligned}$$

   Now by (1ix) and (2i)

$$\begin{aligned} &B \ni K_{AB} \\ &B \mid\equiv \#\left(K_{AB}\right) \\ \therefore\ &B \ni N_A, N_B, T_A, T_B \end{aligned}$$

   B can now construct the response message:

$$\begin{aligned} &B \ni M_1, M_2, M_3, M_4 \\ &B \ni H(K_{AB}, AB_5, (M_1, M_2, M_3, M_4)) \end{aligned}$$

4. Message M_5 received by A

   A now receives a message from B which can only be understood *correctly*, if both A and B have agreed upon K_{AB}.

   A performs some pre–calculations:

$$\begin{aligned} &A \ni K_{AB} \\ \therefore\ &A \ni {N'}_A, N_A, N_B, T_A, T_B \\ &A \ni H(K_{AB}, AB_5, (M_1, M_2, M_3, M_4)) \end{aligned}$$

   Because A has been collecting the messages as well (1ix) and holding all previous messages (2i) and (2ii).

   By (I1) A can decrypt the message

$$\begin{aligned} A \lhd &H(K_{AB}, AB_5, (M_1, M_2, M_3, M_4)) \\ &\therefore \\ A \mid\equiv &H(K_{AB}, AB_5, (M_1, M_2, M_3, M_4)) \\ &\rightsquigarrow B \ni H(K_{AB}, AB_5, (M_1, M_2, M_3, M_4)) \end{aligned}$$

A can now make a series of justified conclusions, by (I3)

$$\begin{aligned} A \mid\equiv B \ni {N'}_B, K_{AB} \\ A \mid\equiv B \ni M_1, M_2, M_3, M_4 \\ A \mid\equiv B \ni N_A, T_A, N_B, T_B \end{aligned}$$

A can now validate the identity of the other party:

$$\begin{array}{ll} A \mid\equiv B \mid\sim (N_B, T_B) & A \mid\equiv B \lhd M_1 \\ A \mid\equiv B \mid\equiv A \mid\sim (N_A, T_A) & A \mid\equiv B \mid\sim M_2 \\ A \mid\equiv B \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B & A \mid\equiv B \mid\sim M_3 \\ A \mid\equiv B \mid\equiv \#(K_{AB}) & A \mid\equiv B \lhd M_4 \end{array} \tag{2iv}$$

It should be clear now why the key K_{AB} is hashed into the hash signature $H(K_{AB}, AB_5, (M_1, M_2, M_3, M_4))$. A hash is only validated by inclusion of a secret and a nonce, see (I3), the key K_{AB} is both.

5. Message M_6 received by B

   By a similar argument to that used for A, it is clear:

$$\begin{array}{ll} B \mid\equiv A \mid\sim {N'}_A & B \mid\equiv A \mid\sim M_1 \\ B \mid\equiv A \mid\equiv B \mid\sim (N_B, T_B) & B \mid\equiv A \lhd M_2 \\ B \mid\equiv A \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B & B \mid\equiv A \lhd M_3 \\ B \mid\equiv A \mid\equiv \#(K_{AB}) & B \mid\equiv A \mid\sim M_4 \end{array} \tag{2v}$$

   Since A could only generate this message if in possession of K_{AB}, B can deduce that A is the party with whom it shares the key and the whole protocol run is current.

5 Summary: Innovations and Possible Attacks

Summary The Transport Level Security handshake protocol is quite ingenious: it lets A send a random message under a public key which is used as an identifying secret shared by the parties before it has been established as such. A challenge and response protocol, the challenge is issued in plain–text and the response returns it as cipher–text, so that the challenger can verify that the responder knows the shared session key. This protocol is effectively a challenge and response protocol with the generated session key, which is created from the secret sent under the public–key.

The protocol is also exemplary in its use of stage identifiers and hash digests. The stage identifiers change the hash digest between messages M_5 and M_6. The hash digests validate the whole protocol run.

Attacks The critical point of the protocol is the transmission of the public–key with which the client should respond with a nonce encrypted under it. The man-in-the-middle attack is well known here: all that need be done is to intercept M_3 and substitute a bogus certificate. The fraud then hinges upons the expectations of A, (1i).

1. Impersonate C and B

   If A is not expecting to use C or B then the attacker, M, can substitute a different certificate for another service: D.

2. Impersonate B

   If A is not expecting to use B but is expecting a certificate issued by C then M can create a service D and attempt to have C issue a certificate for it.

It is quite easy to provide a service that looks like B and appears to be at the address of B this is rather more difficult with a certification authority because the public certification scheme proposed in [ITU89] is based upon the following:

- Certification authorities are well–known in that their addresses and public–keys can be obtained from many sources.
- Certificates contain lists of certification authorities which allow a client to match known certification authorities to those found in the certificates.

Certification authorities currently do nothing other than provide certificates, so all a client obtains from a certificate is some accountability. If defrauded the client can attempt to locate the server who perpetrated the fraud.

Another Useful Feature One of the provisions of TLS is to allow the server to pass to the client another key to use in place of the certificate key. This may be necessary for any of the following reasons:

- The client lacks an implementation to encrypt with the server's public–key.
- The server does not wish to use its public–key.
- Restrictions on key size require that a smaller or larger key must be used.

The client would receive a different public–key but that must be signed under the certificate key for the client to have any faith in it. If the client had chosen to use the alternative key because it lacked an implementation, it would still need to decrypt under the certification authority's key, it is unlikely that the client would be able to do this and not make use of the server's certificate key.

The alternative cryptosystem to system used for certificates is the Diffie–Hellman public–key system[DH77].

6 Other sub-protocols

The protocol described above was the named server protocol, where the server must provide a certificate. There are two other sub–protocols.

6.1 Anonymous Server

In this variant, the server is anonymous and creates a public and private key pair to be used to establish the session key. It would usually use the Diffie–Hellman scheme in this case and would simply send to the client the public key instead of the certificate. This does not weaken the protocol at all, the client and the server will be able to mutually authenticate one another, but the server is unknown to the client and to a certification authority. There is no chain of accountability that could help to locate a fraudulent server.

6.2 Named Client and Server

This variant provides some accountability to the server of the client's identity and it relies upon the client having a certificate. The protocol is the same as the named server protocol, but the server can request a certificate from the client prior to the client sending the pre–master secret. If the client has no certificate it replies by returning no certificates, whereupon the server can take its own action, which may well be to raise an error and not complete the protocol.

7 Summary

TLS, like its predecessor the Secure Socket Layer, SSL, does provide both parties with a mutual belief that the shared session key is a fresh secret. It also, like SSL, can provide the client with some account of the server's Internet location and, unlike its predecessor, it does support mutual authentication certificate exchange. Suffice to say that identities can be securely established—using X.509 certificates—and that a session key can be securely established.

A protocol proof is just a basis for a secure implementation. The software engineering of the authentication protocol has to be considered. An example of such a failure to ensure that a software implementation was invulnerable to attack can be found at [CER98]. The problem with that implementation was that error messages proved to be too informative allowing a sophisticated intruder to recover a session key more quickly than by key trial. Lowe's paper [Low96] has some other implementation attacks.

A Appendix: Gong–Needham–Yong Logic

This is only a cursory introduction to this simple proof system, despite stimulating a great deal of research interest it is relatively unchanged.

A.1 Brief History

Authentication protocols had been developed and discussed, in particular the CCITT X.509 protocol of 1987 [CCI88] was the source for some debate and it was [BAN90], which proved a weakness existed in it. The protocol was extended by [GNY90] and it has been adapted and used in other contexts, [ABLP91]. It may even have been superceded by a calculus that is somewhat less intuitive [AG98]. It is now used principally to illustrate that there is more to secure information that just have believing it to be so, for which see [XZX97] and [Low96].

A.2 The Logic

Notation A.1 (Formulae). Formulae is the name used to refer to a bit–string. Certain useful operations can be applied to bit strings, which are given below. X and Y range over formulae; S over formulae that are secrets and K over formulae that are keys.

(X, Y) Resulting bit–string is a concatenation of two formalue.

$\{X\}_K$ **and** $\{X\}_K^{-1}$ Results are bit–strings are X encrypted and decrypted under a symmetric cryptosystem with key K, respectively.

$\{X\}_{+K}$ **and** $\{X\}_{-K}$ Bit–string results are X encrypted under the public key and under the private key, respectively, of an asymmetric cryptosystem with public key $+K$ and and private key $-K$.

$H(X)$ Result is X after having been subjected to a one-way function.

$F(X_1, X_2, \cdots, X_n)$ Bit–string is the result after applying the many-to-one function F, which is an invertible and computationally feasible in both directions, to all of $X_1, X_2, \cdots, X_n$.

Notation A.2 (Statements). Statements make an assertion about a property of a formula. P and Q denote principals—clients, agents and servers: X is a message; K a key; S a secret.

$P \lhd X$ P is told the message X.

$P \ni X$ P holds or can obtain the message X.

$P \mid\sim X$ P has once conveyed X.

$P \mid\equiv X$ P believes the message X.

$P \mid\equiv \#(X)$ P believes that X is a fresh statement, not seen before in this run of the protocol. X is often known as a nonce. (Note freshness is a belief relative to a principal).

$P \mid\equiv \phi(X)$ P believes that X is recognizable: P is able to decode X it has a recognizable transfer syntax. (Same note as above).

$P \mid\equiv P \overset{S}{\leftrightarrow} Q$ P believes it shares the secret S with Q.

$P \mid\equiv \overset{+K}{\rightarrow} Q$ P believes that $+K$ is the public key of Q.

$P \mid\equiv Q \mid\Rightarrow C$ P believes that Q has jurisdiction over C.

$P \mid\equiv Q \mid\Rightarrow Q \mid\equiv *$ P believes that Q has jurisdiction over all of beliefs held by P.

$X \rightsquigarrow C$ X is a message expressing the statement C.

C_1, C_2 Conjunction of two statements: C_1 and C_2.

$*X$ The message did not originate from its current location.

Remark A.1 (Epochs). Time is divided into two epoch: past and present. The present is the run of a protocol. The past is all other runs of protocols. $P \mid\equiv X$, if a pre–condition, is valid for all of the present. Beliefs held in the past are not necessarily carried forward to the present.

Remark A.2 (Encryption). There are some assumptions about encrypted messages:

1. Messages are assumed to be encrypted as a whole.
2. For recipients: each encrypted message contains enough redundancy to allow the recipient to determine, on decryption, that the right key has been used to do so.
3. For senders: each message contains enough information to allow senders to detect and ignore messages that originated from them.

Also,

1. The key cannot be deduced from the encrypted message.
2. The message can be understood by only those who possess the correct decrypting key.

The logic of authentication is a set of inference rules. The premisses are stated above the deductive line, the conclusion below.

Rule A.1 (Universal–Local). This is an axiom from modal logic.

$$\frac{C_1}{C_2} \Rightarrow \frac{P \mid\equiv C_1}{P \mid\equiv C_2} \qquad \text{(Localize)}$$

Rule A.2 (Being–Told). The rules about the "being-told" operator $\lhd$.

$$\frac{P \lhd *X}{P \lhd X} \tag{T1}$$

(T1) says that if one is in receipt of a message that did originate from elsewhere, one is still aware of it.

$$\frac{P \lhd (X, Y)}{P \lhd X} \tag{T2}$$

(T2) says that if one is in receipt of a composite message one is in receipt of each part of it.

$$\frac{P \lhd \{X\}_K, P \ni K}{P \lhd X} \tag{T3}$$

(T3) is simple one must possess the right key to understand encrypted messages.

$$\frac{P \lhd \{X\}_{+K}, P \ni -K}{P \lhd X} \tag{T4}$$

(T4) is the same statement for public–key systems, decrypting with the private key.

$$\frac{P \lhd F(X, Y), P \ni X}{P \lhd Y} \tag{T5}$$

(T5) is the same statement for a combination function.

$$\frac{\{\{X\}_{-K}\}_{+K} = X, P \lhd \{X\}_{-K}, P \ni +K}{P \lhd X} \tag{T6}$$

(T6) is the same statement for public–key systems, but decrypting with the public key. Note with this rule, the requirement that encryption with a public–key and then with the private–key yields the original message. Not all asymmetric cryptosystems have this property.

Rule A.3 (Possession). Rules for the $\ni$ operator:

$$\frac{P \lhd X}{P \ni X} \tag{P1}$$

(P1) states that one can possess what one is told.

$$\frac{P \ni X, P \ni Y}{P \ni (X, Y), P \ni F(X, Y)} \tag{P2}$$

(P2) states that if in possession of two messages one can create a concatenation and apply a function to them.

$$\frac{P \ni (X, Y)}{P \ni X} \tag{P3}$$

(P3) possession of a composite yields possession of the components.

$$\frac{P \ni X}{P \ni H(X)} \tag{P4}$$

(P4) possession of a message allows the hash of it to be generated.

$$\frac{P \ni F(X, Y), P \ni X}{P \ni Y} \tag{P5}$$

(P5) for the combination function $F()$, given X, Y can be determined.

$$\frac{P \ni K, P \ni X}{P \ni \{X\}_K, P \ni \{X\}_K^{-1}} \tag{P6}$$

(P6) one can encrypte and decrypt with key and message.

$$\frac{P \ni +K, P \ni X}{P \ni \{X\}_{+K}} \tag{P7}$$

(P7) encryption under public–key.

$$\frac{P \ni -K, P \ni X}{P \ni \{X\}_{-K}} \tag{P8}$$

(P8) encryption under private–key.

Rule A.4 (Freshness). These rules specify what can be deduced from fresh messages.

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y), P \mid\equiv \#(F(X))} \tag{F1}$$

$$\frac{P \mid\equiv \#(X), P \ni K}{P \mid\equiv \#(\{X\}_K), P \mid\equiv \#\left(\{X\}_K^{-1}\right)} \tag{F2}$$

$$\frac{P \mid\equiv \#(X), P \ni +K}{P \mid\equiv \#(\{X\}_{+K})} \tag{F3}$$

$$\frac{P \mid\equiv \#(X), P \ni -K}{P \mid\equiv \#(\{X\}_{-K})} \tag{F4}$$

$$\frac{P \mid\equiv \#\left(+K\right)}{P \mid\equiv \#\left(-K\right)} \tag{F5}$$

$$\frac{P \mid\equiv \#\left(-K\right)}{P \mid\equiv \#\left(+K\right)} \tag{F6}$$

$$\frac{P \mid\equiv \phi\left(X\right), P \mid\equiv \#\left(K\right), P \ni K}{P \mid\equiv \#\left(\{X\}_K\right), P \mid\equiv \#\left(\{X\}_K^{-1}\right)} \tag{F7}$$

$$\frac{P \mid\equiv \phi\left(X\right), P \mid\equiv \#\left(+K\right), P \ni +K}{P \mid\equiv \#\left(\{X\}_{+K}\right)} \tag{F8}$$

$$\frac{P \mid\equiv \phi\left(X\right), P \mid\equiv \#\left(-K\right), P \ni -K}{P \mid\equiv \#\left(\{X\}_{-K}\right)} \tag{F9}$$

$$\frac{P \mid\equiv \#\left(X\right), P \ni X}{P \mid\equiv \#\left(H(X)\right)} \tag{F10}$$

$$\frac{P \mid\equiv \#\left(H(X)\right), P \ni H(X)}{P \mid\equiv \#\left(X\right)} \tag{F11}$$

Rule A.5 (Recognizability). When one can claim a formula is recognizable.

$$\frac{P \mid\equiv \phi\left(X\right)}{P \mid\equiv \phi\left(X, Y\right), P \mid\equiv F(X)} \tag{R1}$$

$$\frac{P \mid\equiv \phi\left(X\right), P \ni K}{P \mid\equiv \phi\left(\{X\}_K\right), P \mid\equiv \phi\left(\{X\}_K^{-1}\right)} \tag{R2}$$

$$\frac{P \mid\equiv \phi\left(X\right), P \ni +K}{P \mid\equiv \phi\left(\{X\}_{+K}\right)} \tag{R3}$$

$$\frac{P \mid\equiv \phi\left(X\right), P \ni -K}{P \mid\equiv \phi\left(\{X\}_{-K}\right)} \tag{R4}$$

$$\frac{P \mid\equiv \phi\left(X\right), P \ni X}{P \mid\equiv \phi\left(H(X)\right)} \tag{R5}$$

$$\frac{P \ni H(K)}{P \mid\equiv \phi\left(X\right)} \tag{R6}$$

Rule A.6 (Message Interpretation). A secret S used for identification is denoted $\langle S \rangle$—this is to allow it to be distinguished from other secrets that might be in the message.

$$\frac{P \lhd *\{X\}_K, P \ni K, P \mid\equiv P \overset{K}{\leftrightarrow} Q, P \mid\equiv \phi(X), P \mid\equiv \#(X, K)}{P \mid\equiv Q \mid\sim X, P \mid\equiv Q \mid\sim \{X\}_K, P \mid\equiv Q \ni K} \tag{I1}$$

This specifies the flow of beliefs on receipt of an encrypted message under a shared–key cryptosystem. Notice that $P \mid\equiv \#(X, K)$, either the key is fresh or the message is fresh. Usually the message is freshened by adding a nonce (or timestamp).

$$\frac{\begin{array}{ll} P \lhd & *\{X, \langle S \rangle\}_{+K}, \\ P \ni & (-K, S), \\ P \mid\equiv & \overset{+K}{\rightarrow} P, \\ P \mid\equiv & P \overset{S}{\leftrightarrow} Q, \\ P \mid\equiv & \phi(X, S), \\ P \mid\equiv & \#(X, S, +K) \end{array}}{\begin{array}{ll} P \mid\equiv & Q \mid\sim (X, \langle S \rangle), \\ P \mid\equiv & Q \mid\sim \{X, \langle S \rangle\}_{+K}, \\ P \mid\equiv & Q \ni +K \end{array}} \tag{I2}$$

Message sent encrypted under a public–key. Normally such messages are anonymous, since anyone can use the public–key, but an identifying secret S is passed.

$$\frac{P \lhd *H(X, \langle S \rangle), P \ni (X, S), P \mid\equiv P \overset{S}{\leftrightarrow} Q, P \mid\equiv \#(X, S)}{P \mid\equiv Q \mid\sim (X, \langle S \rangle), P \mid\equiv Q \mid\sim H(X, \langle S \rangle)} \tag{I3}$$

Passing a hashed message.

$$\frac{P \lhd \{X\}_{-K}, P \ni +K, P \mid\equiv \overset{+K}{\rightarrow} Q, P \mid\equiv \phi(X)}{P \mid\equiv Q \mid\sim X, P \mid\equiv Q \mid\sim \{X\}_{-K}} \tag{I4}$$

Passing a message encrypted under a private–key does *not* require an identifying secret.

$$\frac{P \lhd \{X\}_{-K}, P \ni +K, P \mid\equiv \overset{+K}{\rightarrow} Q, P \mid\equiv \phi(X), P \mid\equiv \#(X, +K)}{P \mid\equiv Q \ni (-K, X)} \tag{I5}$$

$$\frac{P \mid\equiv Q \mid\sim X, P \mid\equiv \#(X)}{P \mid\equiv Q \ni X} \tag{I6}$$

$$\frac{P \mid\equiv Q \mid\sim (X, Y)}{P \mid\equiv Q \mid\sim X} \tag{I7}$$

Rule A.7 (Jurisdiction). Rules governing the meaning of jurisdiction.

$$\frac{P \mid\equiv Q \mid\Rightarrow C, P \mid\equiv Q \mid\equiv C}{P \mid\equiv C} \tag{J1}$$

$$\frac{P \mid\equiv Q \mid\Rightarrow Q \mid\equiv *, P \mid\equiv Q \mid\sim (X \rightsquigarrow C), P \mid\equiv \#(X)}{P \mid\equiv Q \mid\equiv C} \tag{J2}$$

$$\frac{P \mid\equiv Q \mid\Rightarrow Q \mid\equiv *, P \mid\equiv Q \mid\equiv Q \mid\equiv C}{P \mid\equiv Q \mid\equiv C} \tag{J3}$$

Definition A.1 (Goals of Authentication). For an authentication there are a number of possible goals:

1. Assurance: to assure another principal that a message has been received. $P \mid\equiv Q \lhd X$.
2. Exchanging secrets: to send to another principal a secret:

$$P \mid\equiv P \overset{S}{\leftrightarrow} Q, \quad Q \mid\equiv P \overset{S}{\leftrightarrow} Q$$

3. Exchanging secrets: to send to another principal a secret and be assured of it:

$$P \mid\equiv Q \mid\equiv P \overset{S}{\leftrightarrow} Q, \quad Q \mid\equiv P \mid\equiv P \overset{S}{\leftrightarrow} Q$$

A.3 Example: The Kerberos Protocol

As an illustration of the use of the BAN logic, the Kerberos[NT94] protocol will be analyzed. This protocol is supported within the emerging Transport Level Security specification. It is designed to establish a session key between two principals given a trusted key server.

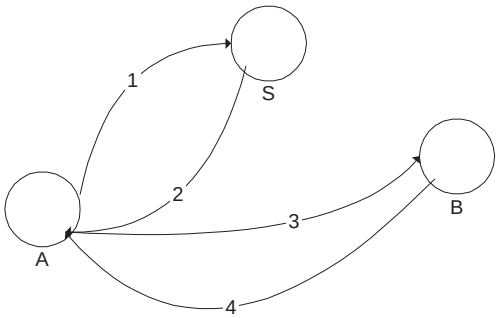

Figure 2: Kerberos Protocol: Messages

Referring to figure 2, the protocol uses three messages to authenticate the sender and the fourth for mutual authentication of the receiver to the sender. The messages appear in order below:

$$A \to S\colon (A, B) \tag{M_1}$$
$$S \to A\colon \{T_S, L, K_{AB}, B, \{T_S, L, K_{AB}, A\}_{K_{BS}}\}_{K_{AS}} \tag{M_2}$$
$$A \to B\colon \{T_S, L, K_{AB}, A\}_{K_{BS}}, \{A, T_A\}_{K_{AB}} \tag{M_3}$$
$$B \to A\colon \{T_A + 1\}_{K_{AB}} \tag{M_4}$$

M_1 A indicates that it wants a session key with B by sending two identifiers to S.

M_2 S returns a message that is encrypted under for A only, it contains a timestamp, a session key, a restatement of the target for which it can be used and, the ingenious part, an encrypted message for B.

M_3 A sends the encrypted part on to B, which contains the timestamp, the session key and the other party to the session key. As well as that, A sends a challenge, encrypted under the session key.

M_4 B responds to the challenge by sending back the timestamp with a predetermined calculation applied to it.

The timestamps T_S and T_A have a lifetime L. Effectively, these act as nonces, so they shall be named as such: N_S and N_A. There are some preconditions about key distributions:

$$A \mid\equiv A \overset{K_{AS}}{\leftrightarrow} S, \quad B \mid\equiv B \overset{K_{BS}}{\leftrightarrow} S \tag{3i}$$

$$S \mid\equiv A \overset{K_{AS}}{\leftrightarrow} S, \quad S \mid\equiv B \overset{K_{BS}}{\leftrightarrow} S \tag{3ii}$$

$$S \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B \tag{3iii}$$

They all hold long–term keys with each S and vice–versa.

$$A \ni K_{AS}, \quad B \ni K_{BS} \tag{4i}$$

$$S \ni K_{AS}, \quad S \ni K_{BS} \tag{4ii}$$

There also some preconditions on jurisdiction and nonces:

$$A \mid\equiv (S \mid\Rightarrow A \overset{K_{AB}}{\leftrightarrow} B), \quad B \mid\equiv (S \mid\Rightarrow A \overset{K_{AB}}{\leftrightarrow} B) \tag{5i}$$

$$A \mid\equiv \#(N_S), \quad B \mid\equiv \#(N_S), \quad B \mid\equiv \#(N_A) \tag{5ii}$$

Finally, the protocol can be analyzed: (M_1) establishes no new beliefs, but with (M_2), the following are established:

$$\begin{aligned} M_2 &\triangleq \{(X,Y)\}_{K_{AS}} \\ X &\triangleq N_S, (A \overset{K_{AB}}{\leftrightarrow} B) \\ Y &\triangleq \{N_S, (A \overset{K_{AB}}{\leftrightarrow} B)\}_{K_{BS}} \\ &\Rightarrow A \mid\equiv S \mid\sim (X,Y) \end{aligned}$$

By initial key distribution, (T3) and (I1)

$$\text{From } X \ \Rightarrow A \lhd (N_S, (A \overset{K_{AB}}{\leftrightarrow} B))$$

By (T3)

$$\Rightarrow A \mid\equiv \# \left(A \overset{K_{AB}}{\leftrightarrow} B\right)$$

By (F1)

$$\Rightarrow A \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B, A \ni K_{AB}$$

By pre–conditions and (J1)

B on receipt of (M_3) can establish the following:

$$M_3 \triangleq (X,Y)$$

where

$$\begin{aligned} X &\triangleq \{N_S, (A \overset{K_{AB}}{\leftrightarrow} B)\}_{K_{BS}} \\ Y &\triangleq \{N_A\}_{K_{AB}} \end{aligned}$$

From X

$$\Rightarrow B \mid\equiv S \mid\sim X$$

By initial key distribution and (I1)

$$\Rightarrow B \lhd (N_S, (A \overset{K_{AB}}{\leftrightarrow} B)$$

By (T3)

$$\Rightarrow B \mid\equiv \# \left(A \overset{K_{AB}}{\leftrightarrow} B\right)$$

By (F1)

$$\Rightarrow B \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B, A \ni K_{AB}$$

By (T1) and jurisdiction pre–conditions

$$\text{From } Y \;\; \Rightarrow B \mid\equiv A \mid\sim N_A$$

By (I1)

$$\Rightarrow B \mid\equiv A \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B, B \mid\equiv A \ni K_{AB}$$

By (I1)

A on receipt of (M_4) can establish the following:

$$M_4 \triangleq \{N_A\}_{K_{AB}}$$

where

$$\Rightarrow A \lhd N_A$$

By (I1)

$$\Rightarrow A \mid\equiv B \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B, A \mid\equiv B \ni K_{AB}$$

By (F1) and (I1)

B Funding and Author Details

Research was funded by the Engineering and Physical Sciences Research Council of the United Kingdom. Thanks to Malcolm Clarke, Russell–Wynn Jones and Robert Thurlby.

Walter Eaves
Department of Electrical Engineering,
Brunel University
Uxbridge,
Middlesex UB8 3PH,
United Kingdom

Walter.Eaves@bigfoot.com
Walter.Eaves@brunel.ac.uk

http://www.bigfoot.com/~Walter.Eaves
http://www.brunel.ac.uk/~eepgwde

References

[ABLP91] M Abadi, M Burrows, B Lampson, and G Plotkin. A calculus for access control in distributed systems. Technical report, DEC Systems Research Center, August 1991. ftp//ftp.digital.com .

[AG98] M Abadi and Andrew D Gordon. A calculus for cryptographic protocols - the spi calculus. Technical report, DEC Systems Research Center, January 1998. ftp//ftp.digital.com .

[BAN90] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, February 1990.

[CCI88] CCITT (Consultative Committee on International Telegraphy and Telephony). *Recommendation X.509: The Directory—Authentication Framework*, 1988.

[CER98] Pkcs. World-Wide Web, July 1998. http://www.cert.org/advisories/CA-98.07.PKCS.html .

[DA97] Tim Dierks and Christopher Allen. The TLS protocol: Version 1.0. In Postel [Pos98], November 1997. ftp://ietf.nri.reston.va.us/internet-drafts/draft-ietf-tls-protocol-05.txt.Z , Expires May 12, 1998.

[DH77] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, V IT-22(6), June 1977.

[DS81] D. E. Denning and M. S. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(7):533–536, August 1981.

[FKK95] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The SSL protocol: Version 3.0. In Postel [Pos98], November 1995. http://www.netscape.com/eng/ssl3/draft302.txt , Expired.

[GNY90] Li Gong, Roger Needham, and Raphael Yahalom. Reasoning about belief in cryptographic protocols. In *Symposium on Research in Security and Privacy*, pages 234–248. IEEE Computer Society, IEEE Computer Society Press, May 1990. http://java.sun.com/people/gong/papers/gny-oakland.ps.gz .

[ITU89] ITU. The directory authentication framework. In *Information technology - Open Systems Interconnection*, number 509 in Series X Recommendations X.200 to X.900. International Telecommunication Union, International Telecommunication Union (ITU), Place des Nations, CH-1211 Geneva 20, Switzerland, 1989. http://info.itu.ch/itudoc/itu-t/rec/x/x200up.html .

[Low96] Gavin Lowe. Some new attacks upon security protocols. In *Proceedings of the Computer Security Foundations Workshop VIII*. IEEE Computer Society Press, 1996.

[NT94] B. Clifford Neuman and Theodore Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications*, 32(9):33–38, September 1994.

[Pos98] Jon Postel, editor. *Internet Requests for Comment Drafts*. Internet Executive Task-Force, http://www.ietf.org , August 1998.

[Ros95] R. Martin Roscheisen. General certificates. World-Wide Web, August 1995. http://www-diglib.stanford.edu/cgi-bin/WP/get/SIDL-WP-1995-0012 , http://www-diglib.stanford.edu/diglib/WP/PUBLIC/DOC18.ps , v 0.9.

[THA99] Thawte digital certificate services. World–Wide Web, Mar 1999. http://www.thawte.com .

[XZX97] Shonhuai Xu, Gendu Zhang, and Hong Xhu. On the properties of cryptographic protocols the weaknesses of ban-like logics. *ACM Operating Systems Review*, 31(4):12–24, October 1997.